C.3 Quantum circuits

1. **Quantum circuit:** A *quantum circuit* is a sequential series of quantum transformations on a quantum register.

2. The inputs are usually computational basis states (all $|0\rangle$ unless stated otherwise).

3. *Quantum circuit diagrams* are drawn with time going from left to right, with the quantum gates crossing one or more “wires” (qubits) as appropriate.
   It represents a sequence of unitary operations on a quantum register rather than physical wires.

4. **Unique features:** Acyclic: loops (feedback) are not allowed. You can apply transforms repeatedly, however.

5. **Fan-In** (equivalent to OR) is not allowed, since it it not reversible or unitary.

6. **Fan-Out** is not allowed, because it would violate the No-cloning Theorem.
   (N.B.: This does not contradict the universality of the Toffoli or Fredkin gates, which are universal only with respect to classical states.)

7. **CNOT:** Fig. III.9 (right) shows the symbol for CNOT and its effect.

8. **Swap:** The swap operation is defined $|xy\rangle \mapsto |yx\rangle$, or explicitly
   \[
   \sum_{x,y\in\{0,1\}} |yx\rangle |xy\rangle.
   \]

9. We can put three CNOTs in series to swap two qubits (Exer. III.29).
   It has a special symbol as shown in Fig. III.11.
¶10. **Controlled-U**: In general, any unitary operator (on any number of qubits) can be controlled (see Fig. III.12). If the control bit is 0, it does nothing, otherwise it does $U$.

¶11. This is implemented by $|0\rangle|0\rangle \otimes I + |1\rangle|1\rangle \otimes U$. Effectively, the *operators* are entangled.

¶12. **Example**: Suppose the control bit is in superposition, $|\chi\rangle = a|0\rangle + b|1\rangle$.

\[
\begin{align*}
( |0\rangle|0\rangle \otimes I + |1\rangle|1\rangle \otimes U ) |\chi, \psi\rangle &= ( |0\rangle|0\rangle \otimes I + |1\rangle|1\rangle \otimes U ) ( a|0\rangle + b|1\rangle ) \otimes |\psi\rangle \\
&= |0\rangle|0\rangle ( a|0\rangle + b|1\rangle ) \otimes I |\psi\rangle + |1\rangle|1\rangle ( a|0\rangle + b|1\rangle ) \otimes U |\psi\rangle \\
&= a |0\rangle \otimes |\psi\rangle + b |1\rangle \otimes U |\psi\rangle \\
&= a |0\rangle |\psi\rangle + b |1\rangle |U \psi\rangle .
\end{align*}
\]

We have a superposition of entangled outputs.

¶13. Recall that CNOT = controlled $X$.

¶14. **Conditional or controlled transformation**: If $U_0$ and $U_1$ are unitary operators, then we can make the choice between them conditional on a control bit as follows:

\[
|0\rangle|0\rangle \otimes U_0 + |1\rangle|1\rangle \otimes U_1.
\]

¶15. For example,

\[\text{CNOT} = |0\rangle|0\rangle \otimes I + |1\rangle|1\rangle \otimes X. \tag{III.17}\]

Figure III.12: Diagram for controlled-$U$ (from NC).
amplitudes. The state input into the circuit where

understood via the equations

26

half of the EPR pair. Amazingly, by doing this he can recover the original state

communication.

good for Alice. Fortunately for Alice, quantum teleportation is a way of utilizing the

the qubit, and moreover can only send

she choose to accept it, is to deliver a qubit

communications channel linking the sender of the quantum state to the recipient.

technique for moving quantum states around, even in the absence of a quantum commu-

trivial, surprising, and a lot of fun – quantum teleportation! Quantum teleportation is a

teleportation. The state to be teleported is

In outline, the steps of the solution are as follows: Alice interacts the qubit

as a

circuit does; it is possible to use quantum circuits to copy classical information encoded

Measurement: It’s also convenient to have a symbol for quantum state measurement, such as Fig. III.14.

| 16. Other special gates: The symbol for the CCNOT gate is show in

or with • for top two connections and ⊕ for bottom, representing

CCNOT|x, y, z⟩ = |x, y, xy ⊕ z⟩,

or put “CCNot” in a box.

| 17. Other operations may be shown by putting a letter or symbol in a box, for example “H” for the Hadamard gate.

| 18. H can be used to generate Bell states (Exer. III.28):

\[ \text{CNOT}(H \otimes I)|xy⟩ = |β_{xy}\rangle. \] (III.18)

| 19. The circuit for generating Bell states (Eq. III.18) is shown in Fig. III.13.

\[ \begin{array}{c|c}
|\psi⟩ & M \\
\hline
|00⟩ & (|00⟩ + |11⟩)/\sqrt{2} \equiv |β_{00}\rangle \\
|01⟩ & (|01⟩ + |10⟩)/\sqrt{2} \equiv |β_{01}\rangle \\
|10⟩ & (|00⟩ − |11⟩)/\sqrt{2} \equiv |β_{10}\rangle \\
|11⟩ & (|01⟩ − |10⟩)/\sqrt{2} \equiv |β_{11}\rangle \\
\end{array} \]

Figure III.13: Quantum circuit for generating Bell states. [from NC fig. 1.12]

\[ \begin{array}{c}
H \\
\hline
M \\
\end{array} \]

Figure III.14: Symbol for measurement of a quantum state (from NC).
C. QUANTUM INFORMATION

C.4 Quantum gate arrays

1. Full adder: Fig. III.15 shows a quantum circuit for a 1-bit full adder.

2. As we will see (Sec. C.7), it is possible to construct reversible quantum gates for any classically computable function. In particular the Fredkin and Toffoli gates are universal.

3. Reversibility: Because quantum computation is a unitary operator, it must be reversible. You know that an irreversible computation \( x \mapsto f(x) \) can be embedded in a reversible computation \( (x, c) \mapsto (g(x), f(x)) \), where \( c \) are suitable constants and \( g(x) \) represents the garbage bits.

Figure III.15: Quantum circuit for 1-bit full adder [from IQC]. “\( x \) and \( y \) are the data bits, \( s \) is their sum (modulo 2), \( c \) is the incoming carry bit, and \( c’ \) is the new carry bit.”

Figure III.16: Quantum gate array for reversible quantum computation.
4. Note that throwing away the garbage bits (dumping them in the environment) will collapse the state (equivalent to measurement) by entangling them in the many degrees of freedom of the environment.

5. Since NOT is reversible, each 1 bit in \( c \) can be replaced by a 0 bit followed by a NOT, so we need only consider \((x, 0) \mapsto (g(x), f(x))\).

6. The garbage must be produced in a standard state independent of \( x \), "because garbage bits whose value depends upon \( x \) will in general destroy the interference properties crucial to quantum computation."

7. **Uncomputation:** This is accomplished by uncomputing. Specifically, perform the computation on four registers \((\text{data}, \text{workspace}, \text{result}, \text{target})\):

\[(x, 0, 0, y) \mapsto (x, g(x), f(x), y).\]

Notice that \( x \) and \( y \) (data and target) are passed through.

8. Now use CNOTs to compute \( y \oplus f(x) \), where \( \oplus \) represents bitwise exclusive-or, in the fourth register:

\[(x, 0, 0, y) \mapsto (x, g(x), f(x), y \oplus f(x)).\]

9. Now we uncompute \( f \), but since the data and target registers are passed through, we get \((x, 0, 0, y \oplus f(x)).\)

Ignoring the result and workspace registers, we write

\[(x, y) \mapsto (x, y \oplus f(x)).\]

10. **Quantum gate array:** Therefore, for any computable \( f : 2^m \to 2^n \), there is a reversible quantum gate array \( U_f : \mathcal{H}^{m+n} \to \mathcal{H}^{m+n} \) such that for \( x \in 2^m \) and \( y \in 2^n \),

\[U_f |x, y\rangle = |x, y \oplus f(x)\rangle,\]

See Fig. III.17.

11. The first \( m \) qubits are called the **data register** and the last \( n \) are called the **target register**.

12. In particular, \( U_f |x, 0\rangle = |x, f(x)\rangle.\)
C. QUANTUM INFORMATION

C.5 Quantum parallelism

\[ U_f \] is linear, if it is applied to a superposition of bit strings, it will produce a superposition of the results of applying \( f \) to them in parallel (i.e., in the same time it takes to compute it on one vector):

\[ U_f(c_1x_1 + c_2x_2 + \cdots + c_kx_k) = c_1U_fx_1 + c_2U_fx_2 + \cdots + c_kU_fx_k. \]

\[ U_f \left( \frac{\sqrt{3}}{2} |x_1\rangle + \frac{1}{2} |x_2\rangle \right) \otimes |0\rangle = \frac{\sqrt{3}}{2} |x_1, f(x_1)\rangle + \frac{1}{2} |x_2, f(x_2)\rangle. \]

\[ U_f \] is linear, if it is applied to a superposition of bit strings, it will produce a superposition of the results of applying \( f \) to them in parallel (i.e., in the same time it takes to compute it on one vector):

\[ U_f(c_1x_1 + c_2x_2 + \cdots + c_kx_k) = c_1U_fx_1 + c_2U_fx_2 + \cdots + c_kU_fx_k. \]

\[ U_f \left( \frac{\sqrt{3}}{2} |x_1\rangle + \frac{1}{2} |x_2\rangle \right) \otimes |0\rangle = \frac{\sqrt{3}}{2} |x_1, f(x_1)\rangle + \frac{1}{2} |x_2, f(x_2)\rangle. \]

\[ W_m|00\ldots0\rangle = \frac{1}{\sqrt{2^m}} (|00\ldots0\rangle + |00\ldots1\rangle + \cdots + |11\ldots1\rangle) \]

\[ = \frac{1}{\sqrt{2^m}} \sum_{x \in 2^m} |x\rangle \]

Figure III.17: Computation of function by quantum gate array [from IQC].
\[
= \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle.
\]

In the last line we are obviously interpreting the bit strings as natural numbers.

¶6. Hence,

\[
U_f W_m |0\rangle = U_f \left( \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x, 0\rangle \right) = \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} U_f |x, 0\rangle = \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x, f(x)\rangle.
\]

¶7. A single circuit does all \(2^m\) computations simultaneously!

¶8. “Note that since \(n\) qubits enable working simultaneously with \(2^n\) states, quantum parallelism circumvents the time/space trade-off of classical parallelism through its ability to provide an exponential amount of computational space in a linear amount of physical space.” [IQC]

¶9. If we measure the input bits, we will get a random value, and the state will be projected into a superposition of the outputs for the inputs we measured.

¶10. If we measure an output bit, we will get a value probabilistically, and a superposition of all the inputs that can produce the measured output.

¶11. Neither of the above is especially useful, so most quantum algorithms transform the state in such a way that the values of interest have a high probability of being measured.

¶12. The other thing we can do is extract common properties of all values of \(f(x)\).

¶13. Both of these require different programming techniques than classical computing.